**Game Lords**

## Description:

* The App builds for the playing html5 games which user having the access code of this application.
* Application contains the great designing and graphics. A user can rename his/her personal information and upload the profile picture.
* A user can play the html5 game on his/her android or ios devices.
* A user can give as favorite and rate particular the html5 game.
* A user can search the game based on the game title name.
* A user can view the game category wise such as Action, Adventure, Premium games, Arcade, and Sports etc.
* A user can view a full description of the html5 games.
* Node js use as a backend for maintaining the data into a database of the application
* To access gamelords application uid is mandatory. Each uid is considered as different user and while accessing site this uid is used to store user related information like user’s favourite games, games liked by user.

**Modules**

Application is divided into number of modules. These modules helps user to find games easily.

1. **HTML5 Games:**

* All HTML5 games are assigned to particular category. Under this section games are listed by category. On game detail page user can view game detail, play game online and view related games.

2. **Favourite**

User can mark games as favourite and Under favourite section he can view all the games marked as favourite.

3. **My Account**

A user can rename his/her personal information and upload the profile picture.

## Mobile Technology:

We are using React Native as the platform to develop the mobile apps for Android, and iOS.

React Native lets you build mobile apps using only JavaScript. It uses the same design as React, letting you compose a rich mobile UI from declarative components.

**Documentation:** [https://facebook.github.io/react-native/docs/0.55/getting-started.html](https://facebook.github.io/react-native/docs/0.40/getting-started.html)

**Version:** 0.55.4

**Folder Structure:**

|  |  |
| --- | --- |
| **android** | Native Android project files |
| **app** | JS files used to create the app logic |
| **ios** | Native iOS project files |

**App Folder Structure:**

|  |  |
| --- | --- |
| **actions** | Actions which defines an object that describes what happened. |
| **assets** | Fonts and Images used in the app |
| **components** | Contains small components that are used in different screens |
| **constants** | Global Size, Base API, Messages |
| **containers** | Screens files |
| **reducers** | Determine how the state will change. |
| **Store** | Shared store configurations |
| **style** | App styles |
| **util** | Other utilities libraries |

**Data Management:**

We use Redux for global app data management.

Redux is one of the hottest libraries for front-end development in today’s marketplace. It is a predictable state container for JavaScript applications and is used for the entire applications state management. Applications developed with Redux are easy to test and can run in different environments showing consistent behavior.

For more info [Click Here](https://redux.js.org/)

**Git:**

**​**Git is used to manage source code versioning.

**Git repo:**

**https://git.sam-media.com/content/gamelord-reactnative.git**

**API List:**

**BASE API URL:** <http://play.game-lords.com>

**Test Token Code:** fdf098fcc6

|  |  |
| --- | --- |
| **Login API** | /api/checkAccess?uid=fdf098fcc6 |

**Usage:** Login API calls when user enter access token and click on continue button.

**Table :** Fetch data from “users” table.

**URL:**

http://play.game-lords.com/api/checkAccess?uid=fdf098fcc6

**Response:**

{

"success": true,

"data": {

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJjZTYyNjFkOC0zMGE4LTRlNzktOGVmOC01MjNiYmQzMTY0MTYiLCJ1aWQiOiJmZGYwOThmY2M2IiwidXNlciI6eyJ1c2VySWQiOiJjZTYyNjFkOC0zMGE4LTRlNzktOGVmOC01MjNiYmQzMTY0MTYiLCJuYW1lIjoiQmFoIiwiZW1haWwiOiJiYWJAZ21haWwuY29tIiwicHJvZmlsZVBpYyI6IjE1MzMxOTgyNjU4NTFjZTYyNjFkOC0zMGE4LTRlNzktOGVmOC01MjNiYmQzMTY0MTZfcHJvZmlsZS5qcGVnIiwiYWRkcmVzcyI6IkJhaCIsImFjY291bnRUeXBlIjoic3RhbmRhcmQiLCJ0aW1lWm9uZSI6IisyOjAwIn0sImlhdCI6MTUzNTM2MjA0OCwiZXhwIjoxNjIxNzYyMDc4fQ.G2p4lOy6J7spVwRYub39qD1YQpP6HYCGbUzuCBEcmpY"

}

}

**Example:**

if (this.state.accessCode !== '') { // Check if the accessCode present in current state or not which user entered in login screen.

const url = vars.BASE\_API\_URL\_GL + `/checkAccess?uid=${this.state.accessCode.toLowerCase()}`; // get bas url and used login API

this.props.show();

axios.get(url) // Calling API

.then(res => {

if (res.data.success === true) { // if success then call redux action checkAccess and store token into redux store

this.props.checkAccess(res.data.data.token);

AsyncStorage.setItem('@AccessToken:key', res.data.data.token); // store token into AsyncStorage

NavigationService.reset("DrawerVOD"); // Redirect to home/welcome screen

} else { // if fails then it displys error

this.setState({ error: 'Please fill valid Access Code' });

this.props.hide(); // Hiding the loader

}

})

} else { // throw the error

this.setState({ error: 'Please fill valid Access Code' });

}

|  |  |
| --- | --- |
| **Get User Profile** | /api/getUserProfile |

**Usage:** Get User Profile API calls when get current authenticated user information including the user’s interests list.

**Table :** Fetch data from “users” and “userInterests” tables, which return combine result of both the tables.

**URL:**

http://play.game-lords.com/api/getUserProfile

**Response:**

{

"success": true,

"data": {

"userId": "ce6261d8-30a8-4e79-8ef8-523bbd316416",

"uid": "fdf098fcc6",

"username": "superadmin",

"password": "$2a$10$Ym9/rlm8.onhyMF/IkoJtuXcvUsIozR2EUIn7CydPCjKG1NVZ82QS",

"name": "Bah",

"email": "bab@gmail.com",

"address": "Bah",

"accountType": "standard",

"profilePic": "1533198265851ce6261d8-30a8-4e79-8ef8-523bbd316416\_profile.jpeg",

"timeZone": "+2:00",

"isActive": "1",

"isDeleted": false,

"createdAt": "2017-12-08T06:50:20.000Z",

"updatedAt": "2018-08-27T09:25:53.000Z",

"userInterests": [

{

"interestId": 1,

"name": "Sports"

},

{

"interestId": 2,

"name": "Action"

},

{

"interestId": 5,

"name": "Entertainment"

}

]

}

}

**Example:**

axios.get(“http://play.game-lords.com/api/getUserProfile”) // calling the API

.then(res => {

if (res.data.success === true) {// if success then call the redux action which store the user info into redux store

console.log( ‘Success’);

this.props.getDetails(userProfile.data.data);

} else { // if fails then display the error while calling api

console.log( 'error’ );

}

})

|  |  |
| --- | --- |
| **Get all html5 Games** | /api/getGames?categoryTypeId=1 |

**Note: categoryTypeId=1 use for HTML5 game.**

**Usage:** Get all html5 games api calls when user want to get the all html5 game from the database and store the all html5 games details into redux store.

**Table :** Fetch data from “games” tables which categoryTypeId is 1, categoryTypeId values 1 because, it returns the games which having category type “HTML5”.

**URL:**

http://play.game-lords.com/api/getGames?categoryTypeId=1

**Response:**

{

"success": true,

"data": [

{

"gameId": 159,

"gameTitle": "Swing Robber",

"gameDesc": "Swing Robber is a HTML5 2D Game..touch to increase the rope and then second touch to swing and go to next platform.Collect dollar bags to make good score.",

"gameImage": "https://cdn.game-lords.com/games/1522226565577\_swing-robber.png",

"gameFile": "https://cdn.game-lords.com/games/1522226565842\_swing-robber/swing-robber/index.html",

"gameReleaseDate": "2018-03-27",

"gameScreenshots": [],

"platforms": "PC, Tablet & Mobile",

"categoryId": 17,

"categoryTypeId": 1,

"userRating": 5,

"categoryName": "Premium Games",

"gameType": "HTML5",

"totalUserReview": 8766,

"isFavorite": false,

"liked": false

},

{

"gameId": 160,

"gameTitle": "NINJA CLAN",

"gameDesc": "Is a fun game with excellent graphics. Many effects and fun sounds. Guarantee hours of fun!",

"gameImage": "https://cdn.game-lords.com/games/1522224853840\_ninja-clan.png",

"gameFile": "1528191244694\_Ninja-Clan/Ninja-Clan",

"gameReleaseDate": "2018-03-27",

"gameScreenshots": [],

"platforms": "PC, Tablet & Mobile",

"categoryId": 17,

"categoryTypeId": 1,

"userRating": 5,

"categoryName": "Premium Games",

"gameType": "HTML5",

"totalUserReview": 6722,

"isFavorite": false,

"liked": false

}, ...

]

}

**Example:**

axios.get(“http://play.game-lords.com/api/getGames?categoryTypeId=1”) // calling api

.then(res => {

if (res.data.success === true) { if success then get the games details which user selected the category and selected associated click on view all in app.

console.log( ‘Success’);

this.props.getGames(games.data.data); // set the games details into redux store.

} else { // if fails then display the error

console.log( 'error’ );

}

})

|  |  |
| --- | --- |
| **Get Game Details** | /api/getGames?gameId=`game\_id` |

**Usage:** Get games api calls when user want to get particular game details.

**Table :** Fetch data from “games” tables which needs to pass particular game id.

**URL:**

http://play.game-lords.com/api/getGames?gameId=156

**Response:**

{

"success": true,

"data": [

{

"gameId": 156,

"gameTitle": "SUPER CARS",

"gameDesc": "Control two cars at the same time. Avoid the obstacles and get the gas tanks.",

"gameImage": "https://cdn.game-lords.com/games/1522223941099\_SuperCars.jpeg",

"gameFile": "https://cdn.game-lords.com/games/1527490297960\_Super-Car/Super-Car/index.html",

"gameReleaseDate": "2018-03-27",

"gameScreenshots": [],

"platforms": "PC, Tablet & Mobile",

"categoryId": 14,

"categoryTypeId": 1,

"userRating": 5,

"categoryName": "Sports",

"gameType": "HTML5",

"totalUserReview": 5766,

"isFavorite": false,

"liked": false

}

]

}

**Example:**

axios.get(“http://play.game-lords.com/api/getGames?gameId=156”) // calliing api and needs to pass game id

.then(res => {

if (res.data.success === true) { if success then game details store into redux store

console.log( ‘Success’);

this.state.getGameDetail(res.data.data); // get game details

} else { // throw the error if fails

console.log( 'error’ );

}

})

|  |  |
| --- | --- |
| **Get Categories** | /api/getCategories?categoryTypeId=1 |

**Usage:** get categories api calls when user wants to get all categories details which category type is 1 it means category is “HTML5”.

**Table :** Fetch data from “categories” tables, which needs to pass categoryTypeId 1. Which indicates and returns only HTML5 categories list.

**URL:**

http://play.game-lords.com/api/getCategories?categoryTypeId=1

**Response:**

{

"success": true,

"data": [

{

"categoryId": 17,

"categoryName": "Premium Games",

"categoryImage": "https://cdn.game-lords.com/games/1522222883523\_hmtl5-premium-small.jpg",

"categoryIcon": "fa fa-star",

"categoryTypeId": 1,

"categoryTypeName": "HTML5"

},

{

"categoryId": 10,

"categoryName": "Action",

"categoryImage": "https://cdn.game-lords.com/games/1513756787016\_html5-action-small.jpg",

"categoryIcon": "fa fa-flash",

"categoryTypeId": 1,

"categoryTypeName": "HTML5"

}, ...

]

}

**Example:**

axios.get(“http://play.game-lords.com/api/getCategories?categoryTypeId=1”) // Calling api and pass categoryTypeId is 1.

.then(res => {

if (res.data.success === true) { if success then it it call the redux function.

console.log( ‘Success’);

this.props.getCategories(categories.data.data); // calling redux function.

} else { // throw error if fails

console.log( 'error’ );

}

})

|  |  |
| --- | --- |
| **Get interests** | /api/interests |

**Usage:** Get interests api calls for the get all interests from the database and store into redux store and it used in account screen

**Table :** Fetch data from “interests” tables and gives the interests list.

**URL:**

http://play.game-lords.com/api/interests

**Response:**

{

"success": true,

"data": [

{

"interestId": 1,

"name": "Sports",

"createdAt": null,

"updatedAt": null

},

{

"interestId": 2,

"name": "Action",

"createdAt": null,

"updatedAt": null

},

]

}

**Example:**

axios.get(“http://play.game-lords.com/api/interests”) // calling api

.then(res => {

if (res.data.success === true) { if success the it calls the redux function

console.log( ‘Success’);

this.props.getInterests(interests.data.data);// calling the redux function and store the all interests list and it’s information into redux store.

} else { // if fails then throw the error

console.log( 'error’ );

}

})

|  |  |
| --- | --- |
| **Get all Favorites of authenticate User** | /api/getFavorites?uid=`user\_id` |

**Usage:** Get all Favorites of authenticate User api calls and used for get the all the list of the favorite games list which is used while display all games list and check which game are favorite or not then set favorite icon accordingly.

**Table :** Fetch data from “favorites” tables and needs to give user id as parameter which defines and returns list of favorite list which user id passed.

**URL:**

http://play.game-lords.com/api/getFavorites?uid=fdf098fcc6

**Response:**

{

"success": true,

"data": [

{

"favoriteId": 7,

"gameId": 27,

"gameTitle": "Demons and Dungeons",

"gameDesc": "A dragon had not been seen in a Thousand years, and the old order of Dragon Slayers had all but disbanded, people lived safely in the knowledge that the greatest threats to their kingdoms and cities was gone forever. That changed the night the sky rained fire. Flaming rock rained all through the world, opening ancient passages to the underworld and from there poured forth a new legion of dragons and their servants. Orcs, goblins, skeletons, ogres, imps, monsters and mad cultists sprung up everywhere, eager to serve new dragon lords and cover the world in ashes. You are the last of the Demon Slayers, and the task rests on you to end the greatest threat the world has ever known, and prevent this from turning into a deadly world of Demons and Dungeons.",

"gameType": "ANDROID",

"gameScreenshots": [],

"gameImage": "https://cdn.game-lords.com/games/1522228583457\_demons-dungeons.png",

"gameFile": "https://cdn.game-lords.com/games/1528880203209\_demons-and-dungeons.apk",

"isFavorite": true,

"gameReleaseDate": "2016-12-31",

"categoryName": "Premium Games",

"platforms": "PC, Tablet & Mobile",

"userRating": 5,

"totalUserReview": 8322

},

{

"favoriteId": 17,

"gameId": 33,

"gameTitle": "Aztec Escape",

"gameDesc": "Aztec Escape is a addictive game with a cool graphic. \r\nUse arrow to move the player to avoid object. Sounds Easy? \r\nBelieve me it is not! You’ll be moving in no time, but it takes super human reflexes to master it! \r\nTry to get as many diamon as possible. \r\nShare your best score with your friend and have fun. I think they will grew jealous of your success. \r\nThe game is fully compatible with all most common mobile devices.",

"gameType": "HTML5",

"gameScreenshots": [],

"gameImage": "https://cdn.game-lords.com/games/1516011778004\_aztec-escape.png",

"gameFile": "https://cdn.game-lords.com/games/1513073545117\_aztec-escape/aztec-escape",

"isFavorite": true,

"gameReleaseDate": "2017-12-12",

"categoryName": "Adventure",

"platforms": "PC, Tablet & Mobile",

"userRating": 5,

"totalUserReview": 7868

}, ...

]

}

**Example:**

axios.get(“http://play.game-lords.com/api/getFavorites?uid=fdf098fcc6”) // calling api

.then(res => {

if (res.data.success === true) { if success then it will call the redux function for store the favorite games list into redux store.

console.log( ‘Success’);

this.props.getFavouriteGames(response.data); // calling redux function and store the data into redux store.

} else { // throw the error if api calling fails

console.log( 'error’ );

}

})

|  |  |
| --- | --- |
| **Set Game As Favorite** | /api/favorite?uid=`user\_id`&gameId=`game\_id`&isFavorite=`boolean` |

**Usage:** Set Game As Favorite calls when user click on the favorite icon. It must be requires the 3 values such as, uid, gameId, and isFavorite.

**Table :** Fetch data from “favorites” tables and needs to give user id, game id, and isFavorite as parameter and request must be **post.**

**URL:**

http://play.game-lords.com/api/favorite?uid=fdf098fcc6&gameId=156&isFavorite=true

Response:

{

"success": false,

"data": {

"success": false,

"data": {}

}

}

**Example:**

axios.post(“http://play.game-lords.com/api/favorite?uid=fdf098fcc6&gameId=156&isFavorite=true”) // calling api and action must be post for this.

.then((response) => {

this.props.showMessage({

message: messages.addToFavorites,

type: true

}); // show success message which is “game set as favorite successfully”

console.log(response);

})

.catch((error) => { // throw the error if fails calling in api

console\_log(error);

});

|  |  |
| --- | --- |
| **Get User Likes Game** | /api/getUserLikes?uid=`user\_id` |

**Usage:** Get User Likes Gameapi used for the get the all user likes game which is used in game details screen and used for check the game is liked by user or not.

**Table :** Fetch data from “userLikes” tables and needs to give user id as parameter.

**URL:**

http://play.game-lords.com/api/getUserLikes?uid=fdf098fcc6

**Response:**

{

"success": true,

"data": [

{

"userLikeId": 2,

"uid": "fdf098fcc6",

"gameId": 8,

"liked": 1,

"gameTitle": "Bear Haven",

"gameDesc": "Bear Haven Nights is a great indie horror game where you have to survive scary nights. \r\n\r\nWelcome to the Bear Haven Motel on you new part time electrician job!\r\n\r\nThere's temporary er... lack of guests at this time, but this shouldn't worry you. Also we have some strange problems in our electrical system at nights, but we're sure you'll manage to fix them.\r\nDon't forget, if there's something broken by 6:00 am you will be fired!\r\nSo don't miss a chance to get a huge payment for your job.\r\nYes... And about our dear guests. Just try not to get in their way.\r\n\r\nBear Haven is a great indie horror game where you have to survive scary nights.\r\nPlease note that it is not yet another \"five nights at freddy's\" clone.\r\nYou don't have just to watch at cameras as a night security guard, but you can freely walk around and interact with the environment.\r\nParticipate in atmospheric and intriguing nightmare adventure.\r\nFix broken electricity and hide from angry bears so they couldn't catch you.\r\n\r\nRun if something goes wrong. Then close the door and turn of the light. The phone shouldn't ring and TV shouldn't work while you are hiding. Then you can survive the first scary night!",

"gameImage": "1516006697621\_Bear-Haven.jpeg",

"gameFile": "1528888943961\_bear-haven.apk",

"gameType": "ANDROID",

"gameScreenshots": [],

"gameReleaseDate": "2017-12-08",

"categoryName": "RPG",

"platforms": "PC, Tablet & Mobile",

"userRating": 3,

"totalUserReview": 1000

},

{

"userLikeId": 5,

"uid": "fdf098fcc6",

"gameId": 6,

"liked": 1,

"gameTitle": "NINJA CLAN",

"gameDesc": "Is a fun game with excellent graphics. Many effects and fun sounds. Guarantee hours of fun!",

"gameImage": "1522224966929\_ninja-clan.png",

"gameFile": "1528191981684\_Ninja-Clan/Ninja-Clan",

"gameType": "HTML5",

"gameScreenshots": [],

"gameReleaseDate": "2017-12-08",

"categoryName": "Action",

"platforms": "PC, Tablet & Mobile",

"userRating": 3,

"totalUserReview": 1000

}, ...

]

}

**Example:**

axios.get(“http://play.game-lords.com/api/getUserLikes?uid=fdf098fcc6”) // calling api

.then((response) => {

console.log(response);

if (response.data.success) { // if success then calling redux function

this.props.getLikesGames(response.data); // calling redux function and store the all likes games into redux store.

}

})

.catch((error) => { // throw the error message

this.setState({ isValid: false, errorMessage: 'Unable to fetch the data.' });

});

|  |  |
| --- | --- |
| **Set Game As Like** | /api/userLike?uid=`user\_id`&gameId=`game\_id`&liked=`boolean` |

**Usage:** Set Game As like calls when user click on the like icon. It must be requires the 3 values such as, uid, gameId, and liked.

**Table :** Fetch data from “userLikes” tables and and needs to give user id, game id, and liked as parameter and request must be **post.**

**URL:**

http://play.game-lords.com/api/userLike?uid=fdf098fcc6&gameId=156&liked=true

**Response:**

{

"success": false,

"data": {

"success": false,

"data": {}

}

}

**Example:**

axios.post(“http://play.game-lords.com/api/userLike?uid=fdf098fcc6&gameId=156&liked=true”) // calling api

.then((response) => {

if (response.status) { // if success the display the success message

this.props.showMessage({

message: messages.addToLikes,

type: true

});

}

})

.catch((error) => { // throw the error

console.log(error);

});

|  |  |
| --- | --- |
| **Upload Profile Pic** | /api/profile/upload?image=`data:image/jpeg;base64` |

**Usage:**  Upload Profile Pic Api used when used upload or edit profile pic in app in account screen. Which requires the image base64 object image as value and it must be post request.

**Table :** Fetch data from “users” tables, returns the user information and need to pass image object as base64

**URL:**

http://play.game-lords.com/api/uploadProfilePic?image=`data:image/jpeg;base64`

**Response:**

{

"success": true,

"data": {

"userId": "ce6261d8-30a8-4e79-8ef8-523bbd316416",

"uid": "fdf098fcc6",

"username": "superadmin",

"password": "$2a$10$Ym9/rlm8.onhyMF/IkoJtuXcvUsIozR2EUIn7CydPCjKG1NVZ82QS",

"name": "Bah",

"email": "bab@gmail.com",

"address": "Bah",

"accountType": "standard",

"profilePic": "1533198265851ce6261d8-30a8-4e79-8ef8-523bbd316416\_profile.jpeg",

"timeZone": "+2:00",

"isActive": "1",

"isDeleted": false,

"createdAt": "2017-12-08T06:50:20.000Z",

"updatedAt": "2018-08-27T09:25:53.000Z",

"userInterests": [

{

"interestId": 1,

"name": "Sports"

},

{

"interestId": 2,

"name": "Action"

},

{

"interestId": 5,

"name": "Entertainment"

}

]

}

}

**Example:**

axios.post(“http://play.game-lords.com/api/uploadProfilePic?image=`data:image/jpeg;base64`”)

.then((response) => {

this.setState({ color: 'green', message: messages.profilePic, showMessage: !this.state.showMessage }) // display the success message on screen

this.props.setProfilePic(response.data.data.profilePic); // calling the redux action and set the response into redux store

this.props.hide(); // hiding the loader on screen

})

.catch((error) => { // throw the error message.

this.props.hide(); // hiding the loader

console\_log(error);

});

|  |  |
| --- | --- |
| **Edit Profile** | /api/editProfile?name=`name`&email=`email`&address=`address`&interests=`interests` |

**Usage:** Edit profile api used when user want to change it’s personal information on account screen.

**Table :** Fetch data from “users” tables, returns the user information and need to pass name, email, address, and interests as array.

**URL:**

http://play.game-lords.com/api/editProfile?name=`name`&email=`email`&address=`address`&interests=`interests`

**Response:**

{

"success": true,

"data": {

"userId": "ce6261d8-30a8-4e79-8ef8-523bbd316416",

"uid": "fdf098fcc6",

"username": "superadmin",

"password": "$2a$10$Ym9/rlm8.onhyMF/IkoJtuXcvUsIozR2EUIn7CydPCjKG1NVZ82QS",

"name": "Bah",

"email": "bab@gmail.com",

"address": "Bah",

"accountType": "standard",

"profilePic": "1533198265851ce6261d8-30a8-4e79-8ef8-523bbd316416\_profile.jpeg",

"timeZone": "+2:00",

"isActive": "1",

"isDeleted": false,

"createdAt": "2017-12-08T06:50:20.000Z",

"updatedAt": "2018-08-27T09:25:53.000Z",

"userInterests": [

{

"interestId": 1,

"name": "Sports"

},

{

"interestId": 2,

"name": "Action"

},

{

"interestId": 5,

"name": "Entertainment"

}

]

}

}

**Example:**

axios.put(“http://play.game-lords.com/api//editProfile?name=`name`&email=`email`&address=`address`&interests=`interests`”) // calling api

.then((response) => {

this.props.showMessage({

message: messages.profileSaved,

type: true

}); // calling the redux action and store the response into redux store

this.props.hide(); // hiding loader screen

this.setState({ color: 'green', message: messages.profileSaved, showMessage: !this.state.showMessage }) // display success message on screen

})

.catch((error) => { // throw error

this.props.hide(); // hiding loader screen

console\_log(error);

console.log(error);

});



**Third Party Packages:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Version** | **Description** |
| **axios** | **0.18.0** | Promise based HTTP client for the browser |
| **react-native-billing** | **2.7.0** | Provide an easy interface to InApp Billing on Android |
| **react-native-device-info** | **0.21.5** | Device Information. |
| **native-base** | **2.3.9** | Essential cross-platform UI components. |
| **react-native-image-picker** | **0.26.7** | Allows you to use native UI to select a photo/video from the device library or directly from the camera. |
| **react-native-in-app-utils** | **6.0.0** | A react-native wrapper for handling in-app payments. |
| **react-native-message-bar** | **2.4.0** | Gradient color. |
| **react-native-orientation** | **3.1.3** | Listen to device orientation changes. |
| **react-native-splash-screen** | **3.1.0** | A splash screen for iOS and Android. |
| **react-native-vector-icons** | **4.5.0** | Customizable Icons. |
| **react-navigation** | **1.4.0** | Routing and navigation. |
| **react-redux** | **5.0.7** | Bindings for Redux. |
| **redux** | **3.7.2** | Predictable state container for JavaScript app. |
| **redux-promise** | **0.5.3** | Promise middleware for Redux. |

**JS File:**

1. **App:**

App is called in index.js to register our application.

We had bind Root Navigator with store to use Redux in our application.

It also contain Navigator Screens and their respective call

1. **SplashScreen:**

This screen is to show the splash image in background till the application get load.

In background a check is done to see the user is logged in or not.

1. **Login:**

For the new user he will be redirected to this screen.

User is asked to add access token for login.

User access-token is stored in Asyncstorage for local storage.

For new user subscription option is provided at the bottom where he can subscribe.

For android - Google play in-app purchase is used. (plugin used: react-native-billing ).

For ios - Apple in-app purchase is used. (InAppUtils from react-native Native Modules ).

API use as follow in this screen,

|  |  |
| --- | --- |
| **Login** | /api/checkAccess?uid=fdf098fcc6 |

1. **Home:**

User can view Banner, Category Slider and List for Games slider based on category wise.

User can set mark as favorite, and view all favorite game list as well. A user can view all category’s games list.

API use as follow in this screen,

|  |  |
| --- | --- |
| **Get User Profile** | /api/getUserProfile |
| **Get all html5 Games** | /api/getGames?categoryTypeId=1 |
| **Get Game Details** | /api/getGames?gameId=`game\_id` |
| **Get Categories** | /api/getCategories?categoryTypeId=1 |
| **Get interests** | /api/interests |
| **Get all Favorites of authenticate User** | /api/getFavorites?uid=`user\_id` |
| **Set Game As Favorite** | /api/favorite?uid=`user\_id`&gameId=`game\_id`&isFavorite=`boolean` |

1. **Favorite:**

User can view his/her own favorite game list. Also, User can set mark as favorite.

API use as follow in this screen,

|  |  |
| --- | --- |
| **Get all Favorites of authenticate User** | /api/getFavorites?uid=`user\_id` |
| **Set Game As Favorite** | /api/favorite?uid=`user\_id`&gameId=`game\_id`&isFavorite=`boolean` |

1. **Category**

User can view game list which category selected.

API use as follow in this screen,

|  |  |
| --- | --- |
| **Get all Favorites of authenticate User** | /api/getFavorites?uid=`user\_id` |
| **Set Game As Favorite** | /api/favorite?uid=`user\_id`&gameId=`game\_id`&isFavorite=`boolean` |

1. **Search:**

It display autosuggest user and allow user to search games.

Search is done locally without api call.On click of search result search text is pushed into *this.props.searchText()* that is used to show data in search screen. User can also set the Set Game as favorite.

API use as follow in this screen,

|  |  |
| --- | --- |
| **Set Game As Favorite** | /api/favorite?uid=`user\_id`&gameId=`game\_id`&isFavorite=`boolean` |

1. **Detail:**

User can get all the information regarding the game. Also, User can play the game on his/her device. User can view the related the game list slider which belongs category

API use as follow in this screen,

|  |  |
| --- | --- |
| **Get all html5 Games** | /api/getGames?categoryTypeId=1 |
| **Get Game Details** | /api/getGames?gameId=`game\_id` |
| **Get all Favorites of authenticate User** | /api/getFavorites?uid=`user\_id` |
| **Set Game As Favorite** | /api/favorite?uid=`user\_id`&gameId=`game\_id`&isFavorite=`boolean` |
| **Get User Likes Game** | /api/getUserLikes?uid=`user\_id` |
| **Set Game As Like** | /api/userLike?uid=`user\_id`&gameId=`game\_id`&isFavorite=`boolean` |

1. **My Account:**

**componentWillMount:** In this API call is done to get user detail using api *getUserProfile.* userDetail is stored in state.

**\_OpenGallery**: This function contain the call to open gallery. User can update profile pic on click of edit profile. *React-native-image-picker* library is used to open gallery & camera from where user can update profile pic.

**submitForm:** function is used to submit user data at the sever end. *editProfile* api is used to edit user information.

API use as follow in this screen,

|  |  |
| --- | --- |
| **Upload Profile Pic** | /api/profile/upload?image=`data:image/jpeg;base64` |
| **Edit Profile** | /api/editProfile?name=`name`&email=`email`&address=`address`&interests=`interests` |

**Redux**:

Redux implementation is done in app. It has all actions inside action folder which also had action type file. Reducer functions are included inside reducer folder. All reducer are combine inside index.js file inside reducer folder.

**Ios certificate:** ios certificate, provisioning file are inside certificate folder.

**Android keystore:** android keystore for generating signed apk is at following path: *‘GameLords/android/app/mobiotv.keystore’*

**Localization:**

For the Localization, You can use **i18n** and **google api** for translation.

1. **I18n:**

Integrates I18n.js with React Native. Uses the user preferred locale as default.

Doing following steps,

1. npm install react-native-i18n --save

## Automatic Setup, react-native link

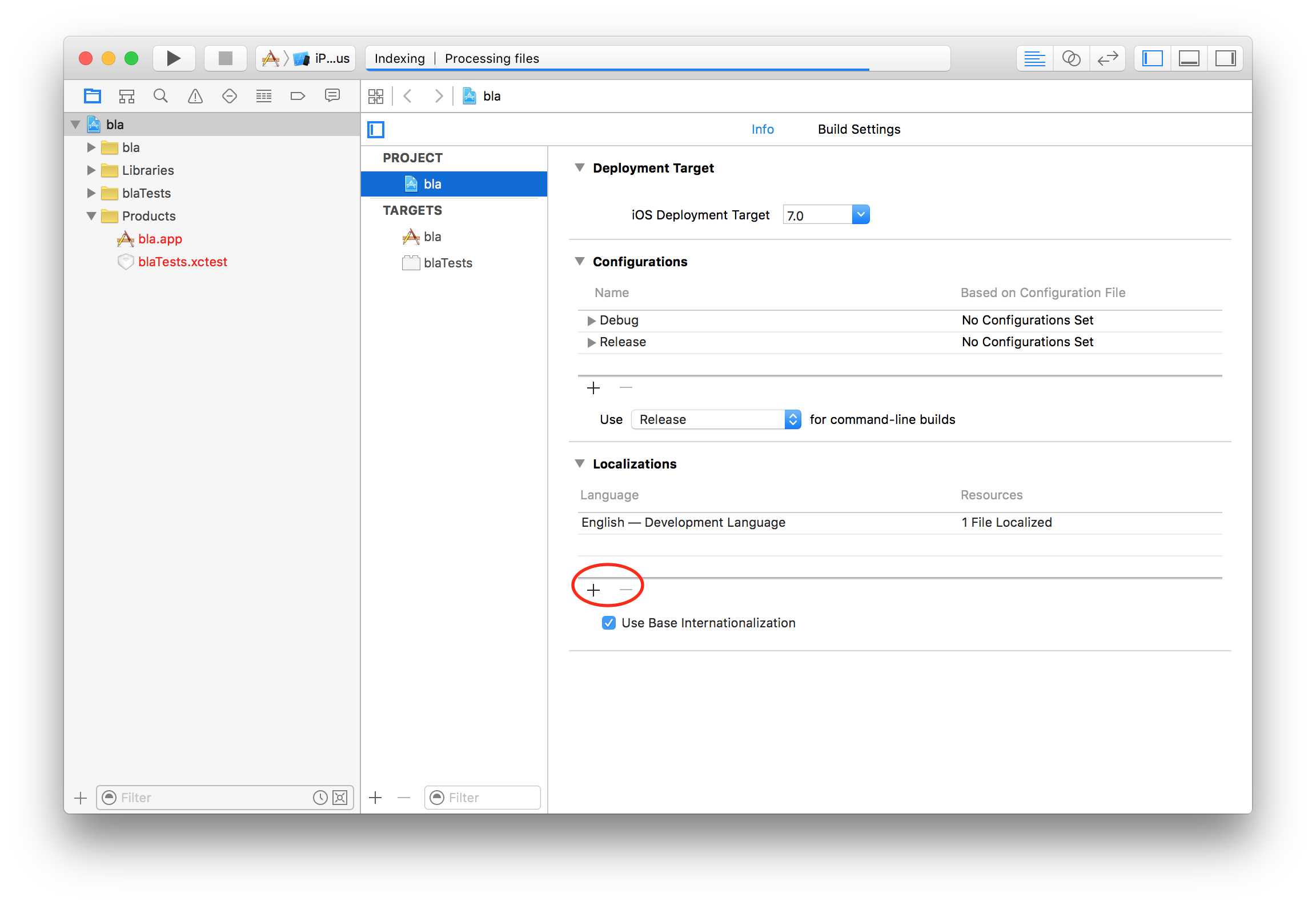
1. Manual SetUp,

### **iOS**

* Add RNI18n.xcodeproj to Libraries and add libRNI18n.a to Link Binary With Libraries under Build Phases.

[More info and screenshots about how to do this is available in the React Native documentation](http://facebook.github.io/react-native/docs/linking-libraries-ios.html#content).

* You also need to add the localizations you intend to support to your iOS project. To do that open your Xcode project:
* $ open <your-project>.xcodeproj

And add the localizations you will support as shown here:

### **Android**

* Add react-native-i18n to your ./android/settings.gradle file as follows:

**include ':app', ':react-native-i18n'**

**project(':react-native-i18n').projectDir = new File(rootProject.projectDir, '../node\_modules/react-native-i18n/android')**

* Include it as dependency in ./android/app/build.gradle file:

**dependencies {**

**// ...**

**compile project(':react-native-i18n')**

**}**

* Finally, you need to add the package to your MainApplication (./android/app/src/main/java/your/bundle/MainApplication.java):

import com.AlexanderZaytsev.RNI18n.RNI18nPackage; // <-- Add to ReactNativeI18n to the imports

// ...

@Override

protected List<ReactPackage> getPackages() {

return Arrays.<ReactPackage>asList(

new MainReactPackage(),

// ...

new RNI18nPackage(), // <-- Add it to the packages list

);

}

// ...

* After that, you will need to recompile your project with react-native run-android.

**Usage**

import I18n from 'react-native-i18n';

// OR const I18n = require('react-native-i18n').default

class Demo extends React.Component {

render() {

return <Text>{I18n.t('greeting')}</Text>;

}

}

// Enable fallbacks if you want `en-US` and `en-GB` to fallback to `en`

I18n.fallbacks = true;

I18n.translations = {

en: {

greeting: 'Hi!',

},

fr: {

greeting: 'Bonjour!',

},

};

For More Info, [react-native-i18n](https://www.npmjs.com/package/react-native-i18n)

1. **Google api:**

A free and unlimited API for Google Translate.

## **Features**

* Auto language detection
* Spelling correction
* Language correction
* Fast and reliable – it uses the same servers that [translate.google.com](https://translate.google.com/) uses

Doing following setps,

npm install --save google-translate-api

**Usage**

From automatic language detection to English:

const translate = require('google-translate-api');

translate('Ik spreek Engels', {to: 'en'}).then(res => {

console.log(res.text);

//=> I speak English

console.log(res.from.language.iso);

//=> nl

}).catch(err => {

console.error(err);

});

**For More Info,** [google-translate-api](https://www.npmjs.com/package/google-translate-api)